On the Markov chain for the move-to-root rule for binary search trees

[short title: Move-to-root rule for binary search trees]

by Robert P. Dobrow and James Allen Fill*

The Johns Hopkins University

September 27, 2004

Abstract

The move-to-root (MTR) heuristic is a self-organizing rule which attempts to keep a binary search tree in near-optimal form. It is a tree analogue of the move-to-front (MTF) scheme for self-organizing lists. Both heuristics can be modeled as Markov chains. We show that the MTR chain can be derived by lumping the MTF chain and give exact formulas for the transition probabilities and stationary distribution for MTR. We also derive the eigenvalues and their multiplicities for MTR.

[1]Research for both authors supported by NSF grant DMS-9311367.

[2]*AMS 1991 subject classifications.* Primary 60J10; secondary 68P10, 68P05.

[3]*Keywords and phrases.* Markov chains, self-organizing search, binary search trees, move-to-root rule, lumping, eigenvalues, simple exchange, move-to-front rule.

1 Introduction and Summary

There has been much interest in recent years in self-organizing search methods. Hester and Hirschberg (1985) survey the field. Hendricks (1989) is a good introduction with numerous applications and open problems.

While most research in this area has been devoted to sequential search techniques for linear lists, a growing body of work addresses heuristics for other data structures. In particular, the binary search tree is a very common and important structure which exploits the ordering of records to achieve faster search time. Records are stored at the nodes of a tree in such a way that a traversal of the tree produces the records in their linear order.

A *binary tree* is a finite tree with at most two "children" for each node and in which each child is distinguished as either a left or right child. By defining an empty binary tree as a binary tree with no nodes we can give a useful recursive definition: a binary tree either is empty or is a node with left and right subtrees, each of which is a binary tree.

Consider a binary tree in which the nodes are labeled with elements of some linearly ordered set. Inorder traversal is a common method for traversing the tree: visit the root after visiting the left subtree and before visiting the right subtree. If this traversal yields the labels in order, the tree is called a *binary search tree.* For example, the set of all binary search trees on 3 nodes is given by:

Figure 1.

Consider a set of n records stored at the nodes of a binary search tree. Assume that record i is accessed with unknown probability p_i and independently of past requests. For simplicity, assume that all the p_i's are strictly positive. As records are accessed we would like to alter the tree dynamically so that the average search cost is made small, where the search cost of a record is defined as one more than the length of the unique path from the root to the node containing the record.

The move-to-root heuristic—described in Section 2—is one self-organizing method which has been studied by several authors. Allen and Munro (1978) introduced the heuristic and gave an exact formula for stationary expected search cost (the asymptotic average cost of retrieving a record). Other treatments of self-organizing trees include Bitner (1979), who considers various search rules, and Sleator and Tarjan (1985), who introduce splay trees and develop (non-probabilistic) amortized analysis of search cost.

This paper is organized as follows: In Section 2 we set notation and describe a many-to-1 mapping between the set of permutations and the set of binary search trees which permits tree operations to be expressed in terms of operations on permutations. We show in Section 3 that the Markov chain for MTR can be obtained by lumping the MTF chain. In Section 4, by exploiting the recursive definition of binary trees and using a simple property of sampling without replacement, we derive the stationary distribution for MTR in a form that is intrinsically tree-based and computationally simple. In Section 5 we give formulas for the k-step transition probabilities, and in Section 6 we analyze the eigenstructure of MTR. In so doing we note interesting parallels with the spectral structure of MTF.

We will treat rates of convergence to stationarity in future work.

2 Notation and preliminaries

Consider an ordered, indexed set of n records. For ease of notation and exposition we identify the records with their indices and just consider $[n] := \{1, 2, \ldots, n\}$ as the set of records.

Let B_n be the set of all labeled binary search trees on n nodes. It can be shown, by exploiting the recursive definition and using generating functions, that $|B_n| = \binom{2n}{n}/(n+1)$. In what follows we use the term "tree" for binary search tree.

Let $\sigma = (\sigma_1, \ldots, \sigma_n) \in S_n$ be a permutation of $[n]$. We will consider σ_k to be the record at the kth position of σ. Define a "tree-building" function $t : S_n \to B_n$ as follows: $t(\sigma)$ is the tree obtained by inserting $\sigma_1, \ldots, \sigma_n$ successively into an empty tree. While technically the function t depends on n, notationally there is no need to distinguish among t for various n.

The function t corresponds to inserting new records into a tree. It is well-defined and onto, and determines an equivalence relation on S_n. We say that two permutations σ and σ' are equivalent if $t(\sigma) = t(\sigma')$, that is, if they correspond to the same tree in the tree-building operation.

Let $\Pi : B_n \to 2^{S_n}$ be the set-valued inverse of t. That is, $\Pi(T) = \{\sigma \in S_n : t(\sigma) = T\}$. The $\Pi(T)$'s are the equivalence classes of S_n.

Note that some authors have considered 1-to-1 mappings between the symmetric group and the space of binary trees in a way that gives a method for ordering and ranking trees. See, for instance, Ruskey and Hu (1977) and Trojanowski (1978). By contrast, here we are considering the set of *all* permutations which can be identified with a particular tree.

The *move-to-root* (MTR) operation is defined as a series of simple exchanges between nodes. A *simple exchange* (SE) for a requested record j is as follows:

(i) Do nothing if j is the root.

(ii) If j is the left child of its parent m, the resulting tree will be the same as the original except for the subtree whose root was m. Record j is "rotated" up to m so that j becomes the root of this subtree. The old left subtree of j doesn't change in relation to j. The old right subtree of j becomes the left subtree of m. The old right subtree of m keeps its relation to m. The transformation is best understood by examining Figure 2-L.

(iii) If j is the right child of m, perform the analogous transformation. (See Figure 2-R.)

The MTR operation performs a sequence of simple exchanges until the requested record is moved to the root of the tree.

Thus MTR and SE are natural analogues of the move-to-front (MTF) and transposition (TR) rules for linear lists. In MTF, an accessed record is brought to the top of the list. In TR, it is transposed with its immediate predecessor.

Figure 2.

For $i \neq j$, we say that i is an *ancestor* of j in T, and write $i <_a^T j$, if j is an element of the subtree which has i as its root. We will suppress the superscript if it is obvious to what tree we are referring. Note that $<_a$ defines a partial order on the nodes of T. A tree is uniquely determined by its ancestry relations and, as the next lemma implies, among trees with the same number of nodes the set of ancestry relations for one tree can never be a proper subset of those for another.

Lemma 2.1 *Let $S, T \in B_n$. Then $T = S$ if and only if $i <_a^T j$ implies $i <_a^S j$ for all $i, j \in [n]$.*

Proof Necessity is trivial; sufficiency follows by a simple induction on n using the recursive definition of a tree. ■

3 Main result: lumping

The mappings t and Π between S_n and B_n make it easy to translate tree operations into operations on permutations. In fact we will show (Lemma 3.4) that MTR for a binary search tree T corresponds to MTF for all of the permutations in $\Pi(T)$.

For n-node trees it is easily shown that the sequence of operations generated by MTR gives an ergodic (aperiodic, irreducible, and positive recurrent) Markov chain on the space B_n.

In the case $n = 3$, the transition matrix for MTR corresponding to the trees in Figure 1 is

$$Q = \begin{array}{c|ccccc} & T_1 & T_2 & T_3 & T_4 & T_5 \\ T_1 & p_1 & 0 & p_2 & p_3 & 0 \\ T_2 & 0 & p_1 & p_2 & p_3 & 0 \\ T_3 & p_1 & 0 & p_2 & 0 & p_3 \\ T_4 & 0 & p_1 & p_2 & p_3 & 0 \\ T_5 & 0 & p_1 & p_2 & 0 & p_3 \end{array}$$

The correspondence between trees and permutations makes it possible to read off the exact transition probabilities for the Markov chain for trees from those for MTF.

Theorem 1 *Let Q be the $|B_n| \times |B_n|$ transition matrix for MTR and let P be the $n! \times n!$ transition matrix for MTF. Then for $S, T \in B_n$,*

$$Q(S,T) = \sum_{\sigma \in \Pi(T)} P(\pi, \sigma),$$

where π is any permutation in $\Pi(S)$.

The theorem is equivalent to the statement that the Markov chain corresponding to P is lumpable (see Kemeny and Snell (1965)) with respect to the map t. From the properties of lumpable chains the following corollaries are immediate:

Corollary 3.1

$$Q^k(S,T) = \sum_{\sigma \in \Pi(T)} P^k(\pi, \sigma), \tag{1}$$

for each $k \geq 0$, where π is any permutation in $\Pi(S)$.

Corollary 3.2

$$Q^\infty(T) = \sum_{\tau \in \Pi(T)} P^\infty(\tau), \tag{2}$$

where P^∞ and Q^∞ are the stationary distributions for the MTF and MTR chains, respectively.

Remark: It is easily seen (e.g., from the case $n = 3$) that the Markov chain corresponding to the simple exchange heuristic is not lumpable with respect to the map t. In addition, for general weights the chain is not time-reversible, unlike the chain corresponding to the transposition heuristic for lists. When all the weights are identical ($p_i \equiv 1/n$), the SE transition matrix is symmetric and so the stationary distribution is uniform on B_n. However, for general weights the stationary distribution—not to mention the k-step transition probabilities and the spectral structure of the transition matrix—is unknown.

The proof of Theorem 1 is based on several observations and lemmas, which follow. A key result is Lemma 3.2 in Allen and Munro (1978), which we reproduce:

Lemma 3.1 *Suppose records i and j have been requested at least once each in a tree modified according to MTR. Let $i < j$. Then $i <_a j$ if and only if the most recent request for i has occurred since the most recent request for any of $i+1, \ldots, j$. Similarly, $j <_a i$ if and only if the most recent request for j has occurred since the most recent request for any of $i, \ldots, j-1$.*

Proof When either simple exchange or MTR is used, if i is requested then i is the only record which *becomes* an ancestor of any records. Also, i will *cease* to be an ancestor of j if and only if an element k, where $i < k \leq j$, is requested. This gives the first part of the lemma. The second part is shown similarly. ∎

For the remaining proofs in this section, as in the preceding proof, the condition that $i > j$ can be handled in the same fashion as the case $i < j$, so we will tacitly restrict ourselves to the latter.

The binary search tree obtained from some permutation σ by the tree-building process is also, as shown by Corollary 3.3 below, the tree obtained from any other binary search tree by successively requesting records in the reverse order of σ. In this regard, note that any binary search tree can be obtained from any other in at most n operations. (That n steps might be necessary is shown by considering the "degenerate" trees corresponding to the identity and reversal permutations.)

Lemma 3.2 *Let $S, T \in B_n$. Consider the following sequence of operations: Choose a leaf in T and within S move the corresponding record to the root. After the record has been moved in S, delete it from T by eliminating its node and the incident branch. Continue until T is empty. Then, after any such sequence of n moves applied to S, the transformed tree will be identical to T before the operation.*

Proof Let S' be the tree obtained from S after the n moves. If $i <_a^T j$ then any k such that $i < k < j$ will be in the subtree of T with root i. Thus the request for i will come after the requests for $i+1, \ldots, j$ because only then will i become a leaf. Thus $i <_a^{S'} j$ by Lemma 3.1. The result now follows from Lemma 2.1. ∎

Corollary 3.3 *Given $\sigma \in S_n$, $t(\sigma)$ is the tree obtained from any n-node tree after making the sequence of requests $\sigma_n, \sigma_{n-1}, \ldots, \sigma_1$.*

Proof For any k, after deleting records $\sigma_n, \ldots, \sigma_{k+1}$ from $t(\sigma)$, σ_k will be a leaf. ∎

This gives a characterization of $\Pi(T)$. For a set S, we use the notation $a < S$ to mean that a is less than every element of S, i.e., that $a < \min S$.

Lemma 3.3 *For $T \in B_n$,*

$$\Pi(T) = \{\sigma : i <_a^T j \;\;\Leftrightarrow\;\; \begin{array}{l} \sigma^{-1}(i) < \{\sigma^{-1}(i+1), \ldots, \sigma^{-1}(j)\}, \; \textit{for } i < j, \\ \sigma^{-1}(i) < \{\sigma^{-1}(j), \ldots, \sigma^{-1}(i-1)\}, \; \textit{for } i > j\}. \end{array}$$

In words, $\Pi(T)$ is the set of all permutations σ such that i is a T-ancestor of j if and only if i is to the left of $i+1, \ldots, j$ when $i < j$ and to the left of $j, \ldots, i-1$ when $i > j$.

Proof The proof, after sorting through the notation, is a direct consequence of the above lemmas. Call the set on the righthand side $\Pi'(T)$. Let $\sigma \in \Pi(T)$. Then $t(\sigma) = T$, and so, by Corollary 3.3, T can be obtained from any tree by requesting $\sigma_n, \ldots, \sigma_1$. Suppose $i <_a^T j$. Then, by Lemma 3.1, i is requested after $i+1, \ldots, j$, so $\sigma^{-1}(i) < \{\sigma^{-1}(i+1), \ldots, \sigma^{-1}(j)\}$. Similarly, the converse holds, showing that $\Pi(T)$ is contained in $\Pi'(T)$.

Suppose $\sigma \notin \Pi(T)$. Then there exist some i and j such that $i <_a^T j$ but $i \not<_a^{t(\sigma)} j$. Since $\sigma \in \Pi(t(\sigma))$, by the first part of the proof $\sigma^{-1}(i) \not< \{\sigma^{-1}(i+1), \ldots, \sigma^{-1}(j)\}$. Hence $\sigma \notin \Pi'(T)$. ∎

A direct consequence is that MTF for permutations corresponds to MTR for the associated binary search trees.

Lemma 3.4 *Fix $\sigma \in S_n$ and let $\sigma' \in S_n$ be the permutation obtained by moving σ_k to the front. Then $t(\sigma')$ is the tree obtained from $t(\sigma)$ by moving σ_k to the root.*

Proof By assumption, $\sigma' = (\sigma_k \sigma_1 \cdots \sigma_{k-1} \sigma_{k+1} \cdots \sigma_n)$. By Corollary 3.3, $t(\sigma')$ is obtained by requesting $\sigma_n, \ldots, \sigma_{k+1}, \sigma_{k-1}, \ldots, \sigma_1, \sigma_k$. But by Lemma 3.1, $t(\sigma')$ is also gotten by requesting $\sigma_n, \ldots, \sigma_1, \sigma_k$. On the other hand, $t(\sigma)$

is obtained by requesting $\sigma_n, \ldots, \sigma_1$. Thus $t(\sigma')$ is obtained from $t(\sigma)$ by moving σ_k to the root. ■

The reader may consult Figure 3 for an illustration of Lemma 3.4. Note that we can read off the ancestry relations of a tree from any one of its associated permutations by looking at the ordering relations. For instance, for $\sigma = (3, 6, 1, 4, 2, 5)$, 3 is an ancestor of everyone; 6 is an ancestor of 4 and 5, but not of 3, 2, or 1 (since 6 is to the left of 4 and 5 but not to the left of 3); 1 is an ancestor of 2, but of no others; 4 is an ancestor of 5 and of no others. This will be true for all equivalent permutations, which include, for instance, $(3, 1, 2, 6, 4, 5)$ and $(3, 1, 6, 4, 5, 2)$.

Figure 3.

We now prove Theorem 1.

Proof Define root(T) to be the record at the root of T. Note that for any $\sigma, \sigma' \in \Pi(T)$, $\sigma_1 = \sigma'_1 =$root(T), since $t(\sigma) = t(\sigma')$. Thus for any fixed π and $T \in B_n$, $P(\pi, \sigma) = 0$ for all but possibly one $\sigma \in \Pi(T)$. If $P(\pi, \sigma) > 0$, then σ is uniquely determined. In particular, since MTF corresponds to composition with a cycle, $\sigma = \pi \circ (k \cdots 1)$, where $k = \pi^{-1}(\text{root}(T))$.

Let $S, T \in B_n$ and $\pi \in \Pi(S)$. Then

$$\sum_{\tau \in \Pi(T)} P(\pi, \tau) = \begin{cases} P(\pi, \sigma) & \text{if there exists } \sigma \in \Pi(T) \text{ such that } P(\pi, \sigma) > 0 \\ 0 & \text{otherwise.} \end{cases}$$

The theorem will follow if we can show that the righthand side above is the same for all $\pi' \in \Pi(S)$.

Suppose there exists $\sigma \in \Pi(T)$ such that $P(\pi, \sigma) > 0$. Let $k = \pi^{-1}(\sigma_1)$. Thus $P(\pi, \sigma) = p_{\pi_k} = p_{\sigma_1}$ and σ is just what results from π after moving record π_k to the front. Let $\pi' \in \Pi(S)$ and $k' = \pi'^{-1}(\pi_k)$. Put $\sigma' = \pi' \circ (k' \cdots 1)$. Thus σ' is just what results from π' after moving $\pi'_{k'} = \pi_k$ to the front and

$P(\pi', \sigma') = p_{\pi'_{k'}} = p_{\pi_k}$. By the previous lemma, $t(\sigma')$ is identical to the tree obtained by moving $\pi'_{k'} = \pi_k$ to the root in S. But since $\pi \in \Pi(S)$, it follows that $t(\sigma)$ is identical to the tree obtained by moving π_k to the root in S. That is, $t(\sigma) = t(\sigma')$, so $\sigma' \in \Pi(T)$ and hence

$$\sum_{\tau \in \Pi(T)} P(\pi', \tau) = P(\pi', \sigma') = P(\pi, \sigma) = p_{\pi_k}.$$

It follows from the foregoing that if $\pi \in \Pi(S)$ and $P(\pi, \sigma) = 0$ for all $\sigma \in \Pi(T)$, then for each $\pi' \in \Pi(S)$, $P(\pi', \sigma) = 0$ for all $\sigma \in \Pi(T)$. ∎

4 Stationary distribution

While Corollary 3.1 gives an exact formula for the transition probabilities of MTR, explicit calculation of these numbers for specific trees is another matter. In the case of the stationary distribution Q^∞, however, exploiting the recursive character of binary trees and using a simple property of sampling without replacement gives a result analogous to that for MTF.

For x a node of a given tree, we use the notation w_x for the probability of accessing the record at that node. For i an index of a given permutation σ, we write w_i for p_{σ_i}.

Theorem 2 *For a tree T,*

$$Q^\infty(T) = \prod_{x \in T} \left(\frac{w_x}{\sum_{y \in T_x} w_y} \right), \tag{3}$$

where T_x is the subtree of T with root x.

Proof For a binary search tree T let $L(T)$ denote the left subtree of T. For a permutation τ let τ^L be the subpermutation of τ induced by the elements of $L(T)$. That is, for $k = 1, \ldots, |L(T)|$, τ^L_k is the kth element of τ which is contained in $L(T)$. Similarly define $R(T)$ and τ^R.

A necessary and sufficient condition for $\tau \in \Pi(T)$ is that the following three conditions hold: (i) τ_1 is the record at the root of T; (ii) $\tau^L \in \Pi(L(T))$; and (iii) $\tau^R \in \Pi(R(T))$.

The stationary distribution for MTF, originally derived by Hendricks (1972), is given by

$$P^{\infty}(\sigma) = \prod_{i=1}^{n} \frac{w_i}{\sum_{j=i}^{n} w_j}.$$

Observe that P^{∞} is the distribution of the order obtained by sampling n items without replacement. It follows from Corollary 3.2 that $Q^{\infty}(T)$ is the probability of sampling n items without replacement in such a way that the first item is at the root of T and the order of choosing the remaining items is consistent with the ancestry relations in $L(T)$ and $R(T)$. Since the root and the two subtrees partition the n items, (3) follows by recursion. ■

As a corollary, we obtain the number of terms in the sum (2).

Corollary 4.1 *For a tree T, let $N(T) = |\Pi(T)|$. Then*

$$N(T) = \binom{|T|-1}{|L(T)|} N(L(T))N(R(T)) = \frac{|T|!}{\prod_{x\in T} |T_x|},$$

where $|T|$ is the number of nodes of T.

Proof The first equation follows from the recursive argument in the proof above. Now iterate to obtain the second equation. ■

Corollary 4.2 *Let T be a nonempty binary search tree. Under MTR if records are accessed uniformly (each with probabily $1/|T|$), then*

$$Q^{\infty}(T) = \frac{1}{\prod_{x\in T} |T_x|}. \quad (4)$$

Remarks:

1. Another way to think about Corollary 4.1 is with respect to partial orderings. The lemma gives the number of linear extensions for a set of elements in a partial order which satisfy some given relations. These relations, of course, must be consistent with the relations satisfied by a binary tree.

2. Computing (3) in linear time requires a simple algorithm which starts at a leaf, working its way up the tree, iteratively computing the partial sum

and partial product for a node after these quantities have been calculated for its children.

3. The distribution (4) arises in the study of random trees. It is the distribution of $t(\sigma)$, where $\sigma \in S_n$ is uniformly distributed. See Mahmoud (1992). The distribution (3) is the distribution of $t(\sigma)$, where $\sigma \in S_n$ has the weighted-sampling-without-replacement stationary distribution of MTF, and so is a generalization of the random permutation model.

4. Unlike the uniform distribution on B_n, the distribution (4) favors trees which are "short and fat." Suppose for ease of di scussion that $n = 2^m - 1$ for integer m. The perfect binary tree is the tree for which all nodes, except for leaves, have 2 children. Call this tree T_m. We can show that the mode of (4) is T_m. It is not hard to derive the asymptotic behavior of $Q^\infty(T_m)$. In particular, the rate of decay for $Q^\infty(T_m)$ is exponential in n. In contrast, $\min_{T \in B_n} Q^\infty(T) = Q^\infty(t(1, \ldots, n)) = 1/n!$ decays at a superexponential rate.

5 Transition probabilities

5.1 A tree-based approach

Our goal in this section is to derive a formula for the k-step transition probabilities $Q^k(S, T)$, where $S, T \in B_n$. The k-step probabilities for MTF were derived by Fill (1993). Thus in light of Corollary 3.1 it would seem that we are done.

Fill's formula, however, is necessarily permutation-based. It depends, for instance, on permutation statistics which are not invariant under the mapping Π. And while the MTF probabilities can be computed in polynomial time, the number of summands in (1) is $N(T)$, which by the pigeonhole principle is, for some T, at least $n!/|B_n| \sim \pi\sqrt{2}\, n^{n+2}(4e)^{-n}$.

The formulas (6) and (8) below have the advantage that they are, at least partially, "tree-based" and can be used to derive numerous characteristics of the chain, including (see Section 6) the eigenvalues and their multiplicities.

Before proceeding to the main theorem of this section (Theorem 3) we establish some notation and preliminary results. It will be necessary to distinguish between the nodes in a tree and the records stored there. Let $R \subseteq [n]$ be a subset of records. For $S, T \in B_n$, define $d(S, T; R)$ to be the indicator of the event that the ancestry relations of the two trees agree for the records in R. That is, $d(S,T;R)=1$ if $i <_a^T j$ exactly when $i <_a^S j$ for all $i,j \in R$, and $d(S,T;R)=0$ otherwise.

For a permutation $\sigma \in S_n$, let $\boldsymbol{\sigma}_m := (\sigma_1,\ldots,\sigma_m)$ for $1 \le m \le n$. Thus $\boldsymbol{\sigma}_m$ is the projection of σ onto its first m coordinates. Recall the definition of $\Pi(T)$ given in Section 2. Let $\Pi_m(T)$ be the projection of the elements of $\Pi(T)$ onto their first m coordinates. Thus $\Pi_n(T)=\Pi(T)$ and $\Pi_1(T)$ is the singleton $\{\mathrm{root}(T)\}$. Finally, let $[\boldsymbol{\sigma}_m]$ denote the *unordered* set $\{\sigma_1,\ldots,\sigma_m\}$ and $\sigma_{(1)} < \cdots < \sigma_{(m)}$ the corresponding order statistics with $\sigma_{(0)} := 0$ and $\sigma_{(m+1)} := n+1$.

An *upset* in a tree $T \in B_n$ is a set U of nodes with the property that if $j \in U$, then the parent (equivalently, all ancestors) of j is in U. Note that the graph in T induced by an upset in T is itself a tree containing (if nonempty) the root of T. We shall refer to this induced tree as the *uptree* U.

It follows from the discussion of the tree-building operation in Section 2 that the uptrees of T consisting of m elements are precisely the trees $t(\boldsymbol{\sigma}_m)$ with $\boldsymbol{\sigma}_m \in \Pi_m(T)$. The discussion in Sections 2 and 3, especially the proof of Lemma 3.1, also yields the following lemma. We leave the simple proof to the reader.

Lemma 5.1 *Consider a sequence Σ of k record requests that contains m distinct records. Suppose that application of Σ to the list $(1,\ldots,n)$ using MTF results in $\boldsymbol{\sigma}_m = (\sigma_1,\ldots,\sigma_m)$ as the m-tuple of frontmost elements. Then application of Σ to a given tree $S \in B_n$ using MTR results in the tree $T \in B_n$ characterized by the following two statements:*

(a) The tree $t(\boldsymbol{\sigma}_m)$ is an uptree of T.

(b) For each $j=0,\ldots,m$, the T-ancestry relations among the records in $(\sigma_{(j)},\sigma_{(j+1)})$ are the same as in S.

Here we use the notation (a,b) for integers a and b to mean the interval of integers strictly between a and b. Note that we take the initial list in Lemma 5.1 to be $(1,\ldots,n)$ only for definiteness. The same result clearly holds for any initial permutation π.

Next we reproduce a result from Fill (1993) concerning MTF:

Lemma 5.2 *Let $P^k(\boldsymbol{\sigma}_m)$ denote the probability, starting in the list $(1,\ldots,n)$, that k requests using MTF move exactly m distinct records to the front and result in $\boldsymbol{\sigma}_m$ as the m-tuple of frontmost elements. Then*

$$P^k(\boldsymbol{\sigma}_m) = w_m^* \sum_{i=0}^{m} (w_i^+)^k w_{m,i},$$

where, for $0 \le i \le m \le n$,

$$w_i^+ := \sum_{h=1}^{i} w_h, \quad w_i^* := \prod_{h=1}^{i} w_h, \quad \textit{and} \quad w_{m,i} := 1/ \prod_{\substack{j \ne i \\ 0 \le j \le m}} (w_i^+ - w_j^+),$$

with the natural conventions $w_0^+ := 0,\ w_0^* := 1,$ *and* $w_{0,0} := 1$.

Proof Noting that the top m records must have their last requests occur in the order $\sigma_m, \sigma_{m-1}, \ldots, \sigma_1$, and conditioning on the times of these requests, we find

$$P^k(\boldsymbol{\sigma}_m) = \sum_{\mathbf{j}_m} (w_m^+)^{j_m} w_m (w_{m-1}^+)^{j_{m-1}} w_{m-1} \cdots (w_1^+)^{j_1} w_1 = w_m^* \sum_{\mathbf{j}_m} \prod_{r=1}^{m} (w_r^+)^{j_r},$$

where the sum is over all m-tuples of nonnegative integers summing to $k - m$. The result follows from an algebraic identity derived in the Appendix of Fill (1993). ■

As discussed in Remark 2.2(a) of Fill (1993), $P^k(\boldsymbol{\sigma}_m)$ can also be written in the form

$$P^k(\boldsymbol{\sigma}_m) = \sum_{i=0}^{m} (w_i^+)^k \, (-1)^{m-i} \, P^\infty(\sigma_1, \ldots, \sigma_i) \, P^\infty(\sigma_m, \sigma_{m-1}, \ldots, \sigma_{i+1}), \quad (5)$$

where $P^\infty(\sigma_1, \ldots, \sigma_i)$ is the probability that sampling without replacement from $[n]$ selects the elements of $\{\sigma_1, \ldots, \sigma_i\}$ in the relative order $(\sigma_1, \ldots, \sigma_i)$; similarly for $P^\infty(\sigma_m, \sigma_{m-1}, \ldots, \sigma_{i+1})$.

The main Theorem 3 now follows directly:

Theorem 3 *Let* $S, T \in B_n$. *Then*

$$Q^k(S,T) = \sum_{m=0}^{n} \sum_{\boldsymbol{\sigma}_m \in \Pi_m(T)} D(S,T;[\boldsymbol{\sigma}_m]) P^k(\boldsymbol{\sigma}_m), \quad (6)$$

where

$$D(S,T;[\boldsymbol{\sigma}_m]) = \prod_{j=0}^{m} d(S,T;(\sigma_{(j)}, \sigma_{(j+1)})).$$

Remarks:

1. From (5) and rearrangement, (6) can alternatively be written as

$$
\begin{aligned}
&Q^k(S,T)\\
&\quad = \sum_{R\subset[n]} (p(R))^k \sum_{m=|R|}^{n} \sum_{\substack{\boldsymbol{\sigma}_m\in\Pi_m(T):\\ [\boldsymbol{\sigma}_{|R|}]=R}} D(S,T;[\boldsymbol{\sigma}_m])\\
&\qquad\qquad \times (-1)^{m-|R|}\, P^\infty(\sigma_1,\ldots,\sigma_{|R|})\, P^\infty(\sigma_m,\sigma_{m-1},\ldots,\sigma_{|R|+1}),
\end{aligned}
\tag{7}
$$

where $p(R) := \sum_{i\in R} p_i$. This form of Q^k will be useful for the spectral analysis of MTR given in Section 6.

2. From (7) we can derive the stationary distribution as given in (2). Let $k\to\infty$ and note that the only term in the outer sum which doesn't vanish is the one corresponding to $R=[n]$. This gives $Q^\infty(T) = \sum_{\sigma\in\Pi(T)} P^\infty(\sigma)$.

3. In the case of equal weights ($p_i \equiv 1/n$),

$$
P^k(\boldsymbol{\sigma}_m) = \sum_{i=0}^{m} \left(\frac{i}{n}\right)^k \frac{(-1)^{m-i}}{i!(m-i)!} =: P^k(m).
$$

Thus

$$
Q^k(S,T) = \sum_{m=0}^{n} P^k(m) C_m(S,T),
$$

where $C_m(S,T) := |\{\boldsymbol{\sigma}_m \in \Pi_m(T) : D(S,T;[\boldsymbol{\sigma}_m]) = 1\}|$.

5.2 Computation of k-step probabilities

While formula (6) is useful for deriving certain characteristics of the MTR chain, we next consider a version that seems better suited for numerical computations. For any tree T, let $\mathrm{rec}(T)$ be the set of records stored at the nodes of T. By rearranging (6) we find that for $S,T\in B_n$,

$$
Q^k(S,T) = \sum_{U\in\mathcal{U}(T)} D(S,T;\mathrm{rec}(U)) Q_k(U), \tag{8}
$$

where $\mathcal{U}(T)$ is the collection of uptrees of T and $Q_k(U) := \sum_{\tau\in\Pi(U)} P^k(\tau)$. Observe that $Q_k(U)$ is the probability that k requests using MTR move exactly $|U|$ distinct records to the root, with these $|U|$ records forming the tree U as a result.

We will derive a recursive (in U) functional relationship for the exponential generating function $\mathcal{Q}_U(z) := \sum_{k=0}^{\infty} Q_k(U) z^k/k!$. Its solution will give a straightforward method for computing the k-step probabilities simultaneously for *all* k. In the remarks at the end of this section we will discuss issues related to the complexity of the calculations.

Theorem 4 *Let U be a binary search tree. Let $\mathcal{Q}_U(z) := \sum_{k=0}^{\infty} Q_k(U) z^k/k!$ be the exponential generating function of the sequence $(Q_k(U))_{k\geq 0}$. Define $\mathcal{Q}_\emptyset(z) := 1$. Then*

$$\mathcal{Q}'_U(z) = w_{\mathrm{root}(U)} e^{w_{\mathrm{root}(U)} z} \mathcal{Q}_{L(U)}(z) \mathcal{Q}_{R(U)}(z), \tag{9}$$

with the initial condition

$$\mathcal{Q}_U(0) = \begin{cases} 1 & \text{if } U = \emptyset \\ 0 & \text{otherwise.} \end{cases}$$

Proof For $k \geq 1$, $Q_k(U)$ is the probability that in k requests: (a) the last request is for $\mathrm{root}(U)$; (b) the request for records in $L(U)$ are such that after the k steps they form $L(U)$; and (c) the request for records in $R(U)$ are such that after the k steps they form $R(U)$. Thus, for $k \geq 1$,

$$Q_k(U) = w_{\mathrm{root}(U)} \sum_{j_1, j_2, j_3} \binom{k-1}{j_1, j_2, j_3} w_{\mathrm{root}(U)}^{j_1} Q_{j_2}(L(U)) Q_{j_3}(R(U)), \tag{10}$$

where the sum is over all non-negative triples which sum to $k-1$. For $k = 0$ it is clear that

$$Q_0(U) = \begin{cases} 1 & \text{if } U = \emptyset \\ 0 & \text{otherwise.} \end{cases}$$

Multiplying both sides of (10) by $z^{k-1}/(k-1)!$ and summing from 1 to ∞ gives the result. ■

We do not know a tree-based closed-form solution to (9) (except in the case of equal weights). The process of solving (9) for $\mathcal{Q}_U(z)$ for all trees U with height at most h is best implemented using a "bottom-up" dynamic programming approach. Beginning with $\mathcal{Q}_\emptyset(z) = 1$, (9) yields $\mathcal{Q}_{\{x\}}(z) = e^{w_x z} - 1$ for a tree $\{x\}$ of height 0. Having computed $\mathcal{Q}_U$ for all trees U with height at most $h-1$, the recursion (9) can be used to find $\mathcal{Q}_U$ for trees U with height h. In each instance, (9) is a first-order linear differential equation involving only linear combinations of exponentials.

For fixed n and $T \in B_n$, the process of solving for $\mathcal{Q}_U$ for all $U \in \mathcal{U}(T)$ can be less tedious. As an illustration, let T be the tree of 3 nodes corresponding to the reversal permutation. (In the notation of Figure 1, $T = T_5$.) The uptrees of T are the empty tree, the singleton tree T' storing 3, the tree T'' induced by records 2 and 3, and T itself. We have

$$\begin{aligned}
\mathcal{Q}_\emptyset(z) &= 1, \\
\mathcal{Q}_{T'}(z) &= e^{p_3 z} - 1, \\
\mathcal{Q}_{T''}(z) &= \frac{p_3}{p_2+p_3}(e^{(p_2+p_3)z} - 1) - (e^{p_3 z} - 1), \\
\mathcal{Q}_T(z) &= \frac{p_2 p_3}{p_1+p_2}(e^z - 1) - \frac{p_3}{p_2+p_3}(e^{(p_2+p_3)z} - 1) + \frac{p_1}{p_1+p_2}(e^{p_3 z} - 1).
\end{aligned}$$

Solving for the coefficients in the generating functions gives, for $k \geq 0$,

$$\begin{aligned}
Q_k(\emptyset) &= \delta_{0k}, \\
Q_k(T') &= p_3^k - \delta_{0k}, \\
Q_k(T'') &= \frac{p_3}{p_2+p_3}((p_2+p_3)^k - \delta_{0k}) - (p_3^k - \delta_{0k}), \\
Q_k(T) &= \frac{p_2 p_3}{p_1+p_2}(1 - \delta_{0k}) - \frac{p_3}{p_2+p_3}((p_2+p_3)^k - \delta_{0k}) + \frac{p_1}{p_1+p_2}(p_3^k - \delta_{0k}),
\end{aligned}$$

where δ_{ij} equals 1 if $i = j$ and 0 otherwise.

Now suppose $S \in B_3$ corresponds to the identity permutation. (In terms of Figure 1, $S = T_1$.) Then

$$\begin{aligned}
D(S, T; \emptyset) &= D(S, T; \{3\}) = 0 \quad \text{and} \\
D(S, T; \{2, 3\}) &= D(S, T; \{1, 2, 3\}) = 1,
\end{aligned}$$

and so

$$\begin{aligned}
Q^k(S, T) &= Q_k(T'') + Q_k(T) \\
&= \begin{cases} \frac{p_2 p_3}{p_1+p_2}(1 - p_3^{k-1}) & \text{if } k \geq 1 \\ 0 & \text{if } k = 0. \end{cases}
\end{aligned}$$

Remarks:

1. When T is "long and skinny"—that is, when T is "close" to the tree obtained by the identity or reversal permutation—$Q^k(S,T)$ can be computed in time polynomial in n using any of the methods we have discussed. For example, one can use (1) and the formula from Fill (1993) for the MTF transition probability $P^k(\pi,\sigma)$. The latter can be computed in polynomial time for fixed $\sigma \in S_n$, and it is not hard to show that if $T \in B_n$ has height $n-1-k$, then $N(T) = |\Pi(T)| \le n^k$.

2. Let $u(T)$ denote the number of uptrees for tree T. Thus $u(T)$ is the number of terms in the sum in (8). Then u satisfies the recursion

$$u(T) = 1 + u(L(T))u(R(T)). \tag{11}$$

For example, for the perfect binary tree on $n = 2^m - 1$ nodes let u_m denote the number of uptrees. Then

$$u_{m+1} = u_m^2 + 1,\ \ m \ge 0, \tag{12}$$

which generates the sequence $1, 2, 5, 26, 677, 458330, 210066388901, \ldots$.

Note that u_m is the number of binary search trees with height at most $m-1$ and (12) has been studied from this point of view. While no closed form solution to (12) is known, one can show that $u_m = \lfloor K^{2^m} \rfloor = \lfloor K^{n+1} \rfloor$ where K is approximately 1.502837. (See Aho and Sloane (1973) for a discussion of this and other nonlinear recurrences of the form $x_{n+1} = x_n^2 + g_n$, where g_n is a slowly growing function of n.)

3. One approach to computing $D(S,T;\cdot)$ begins by constructing tables of ancestry relations for S and T. It is easy to see how to construct such tables in time—and space—$O(n^2)$. By constructing an ancestry table as a hash table, a single ancestry relation can be checked in constant time and thus $D(S,T;R)$ computed in time $O((n-|R|)^2) = O(n^2)$ for fixed S, T, R.

6 Eigenanalysis of MTR

The fact that MTF is lumpable gives us little to go on in trying to determine the eigenstructure of MTR. From lumpability it follows that the eigenvalues for MTR are some subset of those for MTF. But determining *which* subset and the corresponding multiplicities requires more detailed analysis.

Phatarfod (1991) derived the eigenvalues and multiplicities for MTF. Suppose for simplicity throughout this section that sums of distinct collections of weights are distinct. Then the eigenvalues are all the partial sums of the weights, excluding the n cases where the summation is over $n-1$ weights. The multiplicity of each eigenvalue $\lambda_R = \sum_{j \in R} p_j$ corresponding to a sum of $|R| = m$ weights is the number of permutations in S_n fixing exactly those points in R, namely, the number of derangements (permutations with no fixed points) in S_{n-m}.

Our results for MTR exhibit an interesting parallelism to those for MTF. In brief, we shall define the notions of unit gap and fixed point of a tree and show (i) that the eigenvalues for MTR are the partial sums of weights excluding sets which have unit gaps, and (ii) that the multiplicity of the eigenvalue λ_R is the number of trees in B_n fixing exactly those points in R.

For $R \subseteq [n]$, write $r_1 < r_2 < \cdots < r_m$ for the elements of R. Define $r_0 := 0$ and $r_{m+1} := n+1$. Let

$$g_i(R) := r_{i+1} - r_i - 1, \quad i = 0, \ldots, m,$$

denote the number of integers in the interval (r_i, r_{i+1}). Then $g_i(R)$ is called the *i-th gap* of R.

We say that a tree T *fixes* a record j if the records $j+1, \ldots, n$ are all in the right subtree of j and the left subtree of j is empty. Equivalently, T fixes j if there exists $\pi \in \Pi(T)$ such that $\pi(j) = j$ and π maps $\{1, \ldots, j-1\}$ to itself and $\{j+1, \ldots, n\}$ to itself.

We say that a tree fixes a set of records R if the tree fixes each of the records in R. Denote the number of trees which fix exactly R by $\alpha_n(R)$. We call a tree which fixes none of its records a *derangement tree.* Write $\alpha_n := \alpha_n(\emptyset)$ for the number of n-node derangement trees.

Note that if a tree T fixes exactly one record j, then the nodes of T which contain records $1, \ldots, j-1$ form a derangement tree. Similarly, the nodes of T which contain records $j+1, \ldots, n$ also form a derangement tree. Conversely, any derangement tree on $\{1, \ldots, j-1\}$ can be joined with any derangement tree on $\{j+1, \ldots, n\}$ to obtain a tree with j as its unique fixed point.

It now follows by iteration that

$$\alpha_n(R) = \prod_{i=0}^{|R|} \alpha_{g_i(R)}. \tag{13}$$

Similarly, with the conventions $\alpha_0 = \beta_0 = 1$,

Lemma 6.1 *Let $\beta_n = \binom{2n}{n}/(n+1)$ denote the number of binary search trees on n nodes. Then (β_n) satisfies the following recursive relationship with respect to (α_n):*

$$\beta_n = \alpha_n + \sum_{j=1}^{n} \alpha_{j-1}\beta_{n-j}, \quad n \geq 1. \tag{14}$$

Corollary 6.1 *The following formula gives the number of derangement trees on n nodes:*

$$\alpha_n = \frac{1}{2}\left[\left(-\frac{1}{2}\right)^n + \sum_{j=0}^{n}\left(-\frac{1}{2}\right)^j \beta_{n-j}\right], \quad n \geq 0. \tag{15}$$

Proof Recall that the generating function for the nth Catalan number β_n is

$$\mathcal{B}(z) = \frac{1-\sqrt{1-4z}}{2z}.$$

From (14) it follows that the generating function for the number of derangement trees is

$$\mathcal{A}(z) = \frac{\mathcal{B}(z)}{1+z\mathcal{B}(z)} = \frac{1+\mathcal{B}(z)}{2+z}, \tag{16}$$

and the result follows by computing the coefficient of z^n. ■

Remarks:

1. Values of α_n up through $n = 21$ can be found in Sloane (1973), sequence number 635. The first 10 numbers, starting with α_1, are: 0, 1, 2, 6, 18, 57, 186, 622, 2120, 7338.

2. The sequence (α_n) has arisen in the context of Fine's (1970) work on closeness relations. We shall not go into detail about the connections. In

short, α_n is the number of admissible closeness relations (in Fine's terminology) on $[n]$. Fine gave a method of calculating α_n but did not produce an explicit formula like our (15).

3. It is easy to show that α_n satisfies the following recursive relationship with respect to β_n:

$$\alpha_n = \frac{1}{2}(\beta_n - \alpha_{n-1}),\ n \geq 1; \tag{17}$$

furthermore, β_n satisfies

$$\beta_n = \frac{2(2n-1)}{n+1}\beta_{n-1}, \quad n \geq 1. \tag{18}$$

We feel that the simplest method for calculating α_n is to calculate β_n iteratively and then use (17) iteratively to get α_n.

4. Combining (17) and (18) gives a simple recurrence relation satisfied by (α_n):

$$2(n+1)\alpha_n = (7n-5)\alpha_{n-1} + 2(2n-1)\alpha_{n-2}, \quad n \geq 2.$$

We now give a tree-based description of the spectral structure of MTR.

Theorem 5 *The transition matrix for MTR is diagonalizable. The eigenvalues of Q are those values*

$$\lambda_R := p(R) = \sum_{j \in R} p_j$$

for which R has no gaps of size 1. The multiplicity μ_R of λ_R is the number of n-node trees which fix exactly those points in R. That is,

$$\mu_R = \alpha_n(R),$$

which can be computed directly from (13) and the formula for the number of derangement trees.

Proof We identify the eigenvalues and their multiplicities by calculating the trace of Q^k. Consider formula (7). When $S = T$, $d(S, T; R') = 1$ for all $R' \subseteq [n]$ and the coefficient of $(p(R))^k$ simplifies to

$$\sum_{m=|R|}^{n} \sum_{\substack{\boldsymbol{\sigma}_m \in \Pi_m(T): \\ [\boldsymbol{\sigma}_{|R|}] = R}} (-1)^{m-|R|}\, P^\infty(\sigma_1, \ldots, \sigma_{|R|})\, P^\infty(\sigma_m, \sigma_{m-1}, \ldots, \sigma_{|R|+1}).$$

Summing over $T \in B_n$ gives

$$\sum_{m=|R|}^{n} \sum_{\boldsymbol{\sigma}_m : [\boldsymbol{\sigma}_{|R|}]=R} (-1)^{m-|R|} P^{\infty}(\sigma_1, \ldots, \sigma_{|R|}) \, P^{\infty}(\sigma_m, \sigma_{m-1}, \ldots, \sigma_{|R|+1}) \, \tau(\boldsymbol{\sigma}_m), \tag{19}$$

where $\tau(\boldsymbol{\sigma}_m) = |\{T \in B_n : \boldsymbol{\sigma}_m \in \Pi_m(T)\}|$.

It is easily seen that $\tau(\boldsymbol{\sigma}_m)$ depends only on the unordered set $[\boldsymbol{\sigma}_m]$ and equals

$$\tau(\boldsymbol{\sigma}_m) = \prod_{i=0}^{m} \beta_{g_i([\boldsymbol{\sigma}_m])}. \tag{20}$$

Therefore (19) equals

$$\sum_{m=|R|}^{n} (-1)^{m-|R|} \sum_{\substack{U \supseteq R \\ |U|=m}} \tau(U) = \sum_{U \supseteq R} (-1)^{|U|-|R|} \tau(U). \tag{21}$$

But $\tau(U)$ is, by (20), precisely the number of trees that fix *at least* the points in U. By Möbius inversion, (21) reduces to $\alpha_n(R)$. ■

Remarks:

1. For $n \geq 2$, the second largest eigenvalue is the sum which leaves out the consecutive pair $\{i, i+1\}$ with the smallest total weight. Its multiplicity (assuming no ties) is $\alpha_2 = 1$.

2. As in the case of MTF, when the weights are uniform the eigenvalues of MTR are the numbers

$$0,\ 1/n,\ 2/n, \ldots, (n-2)/n,\ 1.$$

The multiplicity of the eigenvalue m/n is the number of trees which fix exactly m points.

7 References

Aho, A. V. and Sloane, N. J. A. (1973). Some doubly exponential sequences. *Fibonacci Quarterly*, **11** 429–437.

Allen, B. and Munro, I. (1978). Self-organizing binary search trees. *J. ACM*, **25** 526–535.

Bitner, J. R. (1979). Heuristics that dynamically organize data structures. *SIAM J. Comp.,* **8** 82–110.

Fill, J. A. (1993). An exact formula for the move-to-front rule for self-organizing lists. Technical Report #529, Department of Mathematical Sciences, The Johns Hopkins University.

Fine, T. (1970). Extrapolation when very little is known about the source. *Info. and Control.*, **16** 331–359.

Hendricks, W. J. (1972). The stationary distribution of an interesting Markov chain. *J. Appl. Probab.*, **9** 231–233.

Hendricks, W. J. (1989). *Self-organizing Markov Chains.* MITRE Corp., McLean, Va.

Hester, J. H. and Hirschberg, D. S. (1985). Self-organizing linear search. *Comp. Surveys,* **17** 295–311.

Kemeny, J. G. and Snell, J. L. (1965). *Finite Markov Chains.* D. Van Nostrand Co., Princeton, N.J.

Knuth, D. E. (1973). *The Art of Computer Programming.* Vols. 1, 3. Addison-Wesley, Reading, Mass.

Mahmoud, H. M. (1992). *Evolution of Random Search Trees.* John Wiley & Sons, Inc., New York.

Phatarfod, R. M. (1991). On the matrix occurring in a linear search problem. *J. Appl. Prob.*, **28** 336–346.

Ruskey, F., and Hu, T. C. (1977). Generating binary trees lexicographically. *SIAM J. Comp.*, **6** 745–758.

Sleator, D. D., and Tarjan, R. E. (1985). Self-adjusting binary search trees. *J. ACM*, **32** 652–686.

Sloane, N. J. A. (1973). *A Handbook of Integer Sequences,* Academic Press, New York.

Trojanowski, A. E. (1978). Ranking and listing algorithms for k-ary trees. *SIAM J. Comp.*, **7** 492–509.

Robert P. Dobrow
Department of Mathematical Sciences
The Johns Hopkins University
Baltimore, MD 21218-2689

James Allen Fill
Department of Mathematical Sciences
The Johns Hopkins University
Baltimore, MD 21218-2689